



HackaBLE

A Bluetooth Low Energy Primer

**Every BLE device is 1 hop away from
a compromised laptop**



whoami

ben smith

- "security researcher"
 - good at packets
 - bad at slides
-
- I like over-the-air stuff
 - I like stuff you can hack with just a phone

Table of Contents

5 What things are we talking about

8 What is BLE?

8 How does it work?

17 Security

21 How Hack?

26 Interesting Vulns

28 A Note About DFU

29 Demo

31 Questions

What stuff are we talking about here?

Everything.

Literally everything.

Processing power in everything.

What uses BLE?

- Sofas
- Beds
- Firearms & Tasers
- Headphones & Hearing Aids
- Locks
- Lights
- Digital Signage & TVs
- Shopper Tracking
- Vehicle & Worker Tracking
- Street Lights
- Speakers
- Point of Sale / SigCap / Receipt Printer
- Printers
- Car Chargers
- Health
 - Diabetes monitors
 - Bikes
- Curtains
- Smoke Alarms
- Fans
- ...
- Everything

What scares me?

- Kinetic effects!
 - KDoS - Kinetic Denial of Service
- Data exfil
- Sensitive data
 - Device Name ("Wei's Beats Pro")
 - Temperature, shopper count, location
- Engine information & faults
- Remotely flashable
- Vendor defaults
- Write enabled



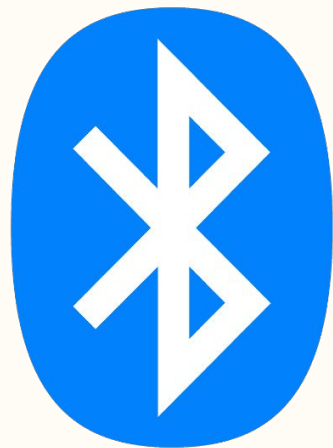
What is BLE?

X Bluetooth Classic (BR/EDR)

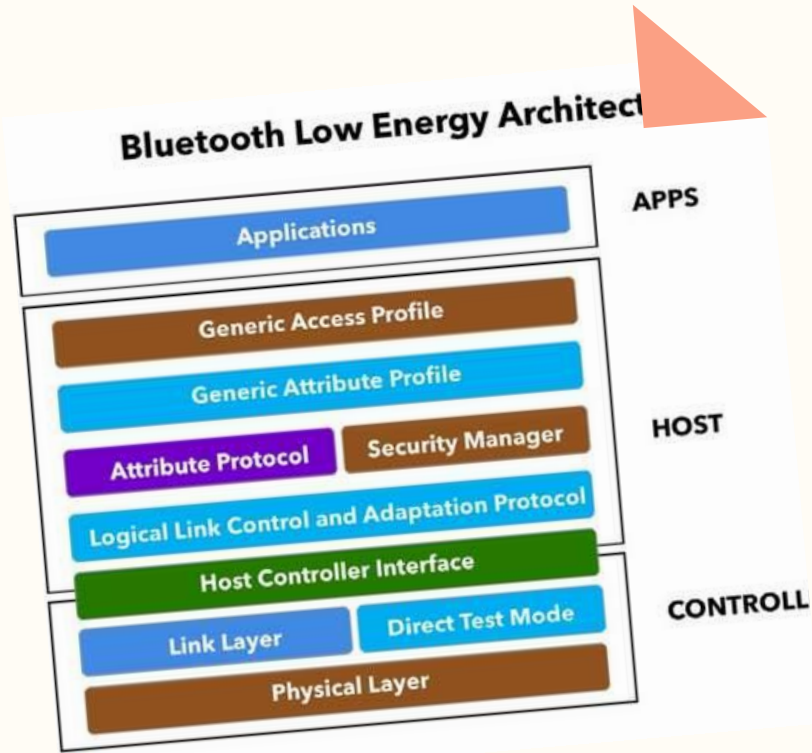
> Bluetooth Low Energy

Bluetooth Low Energy

A whole new protocol



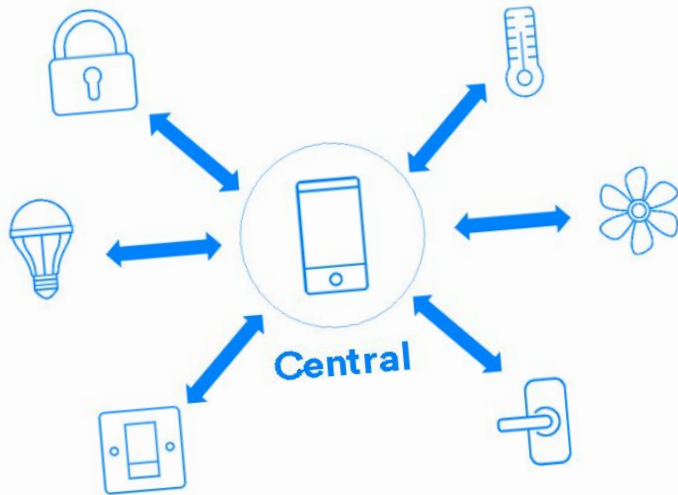
Bluetooth



BLE Stack

HCI - how your laptop talks to the radio

GATT - where the action happens



Roles

Central (cell phone)

Peripheral (IoT device)

Broadcaster

Observer

```

> Frame 2471: 38 bytes on wire (304 bits), 38 bytes captured (304 bits)
> nRF Sniffer for Bluetooth LE
✓ Bluetooth Low Energy Link Layer
  Access Address: 0x8e89bed6
  > Packet Header: 0x0c04 (PDU Type: SCAN_RSP, TxAdd: Public)
    Advertising Address: 66:ea:da:00:26:6f (66:ea:da:00:26:6f)
  ✓ Scan Response Data: 05094c554349
    ✓ Advertising Data
      ✓ Device Name: LUCI
        Length: 5
        Type: Device Name (0x09)
        Device Name: LUCI
  CRC: 0x1e3358

```

Advertising

Active or Passive

Types: Connectable, Scannable, Directable

Addresses:

Public address

Random static address

Resolvable random private address

Non-resolvable random private address

Bluetooth Low Energy Link Layer

Access Address: 0x8e89bed6

> Packet Header: 0x2300 (PDU Type: ADV_IND, ChSel: #1, TxAdd: Public)

Advertising Address: 66:ea:da:00:26:6f (66:ea:da:00:26:6f)

✓ Advertising Data

> Flags

✓ 128-bit Service Class UUIDs

Length: 17

Type: 128-bit Service Class UUIDs (0x07)

Custom UUID: e4490001-60c7-4baa-818d-235695a2757f (Unknown)

✓ 16-bit Service Class UUIDs

Length: 7

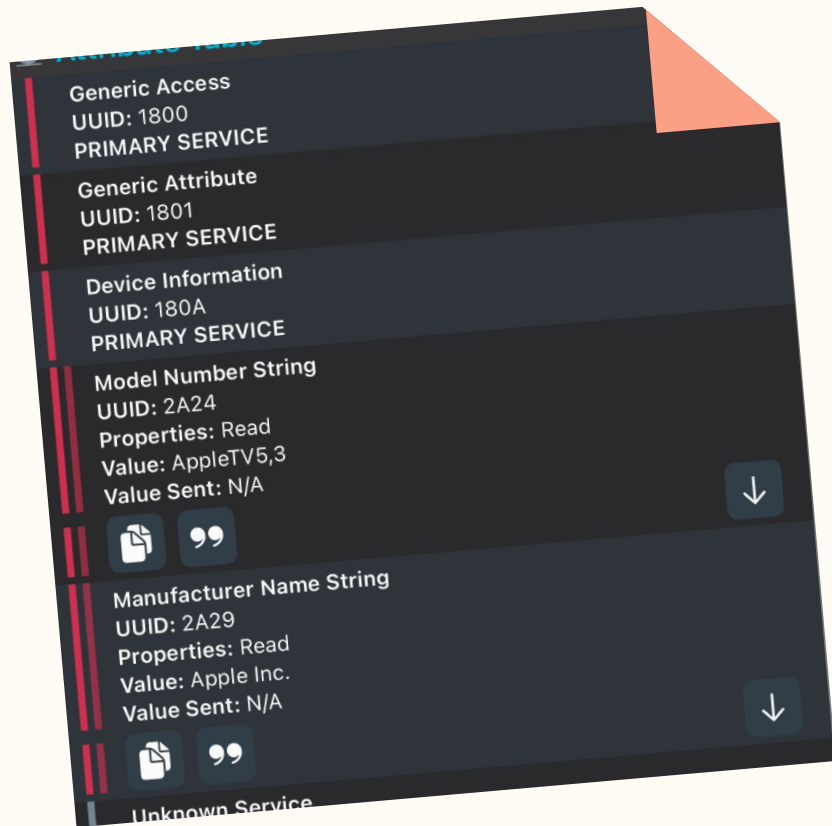
Type: 16-bit Service Class UUIDs (0x03)

UUID 16: Dialog Semiconductor GmbH (0xfef5)

UUID 16: Battery Service (0x180f)

UUID 16: Current Time Service (0x1805)

CRC: 0x5c3d29



Generic Attribute Profile (GATT)

Services

Characteristics

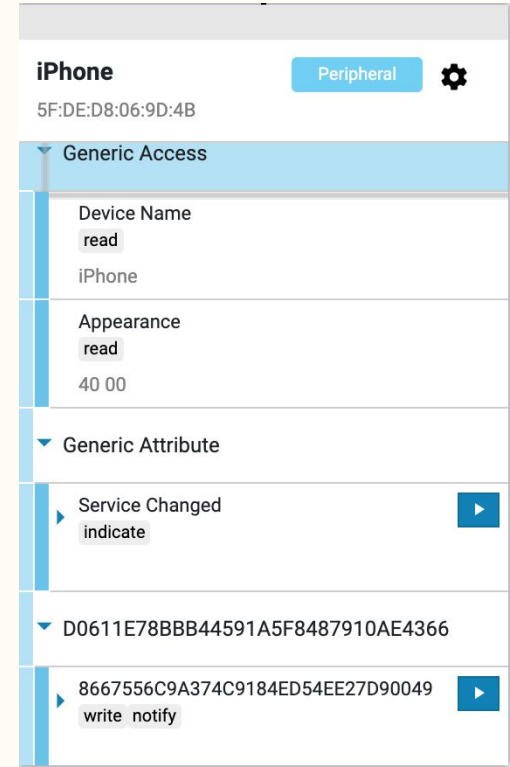
Descriptors

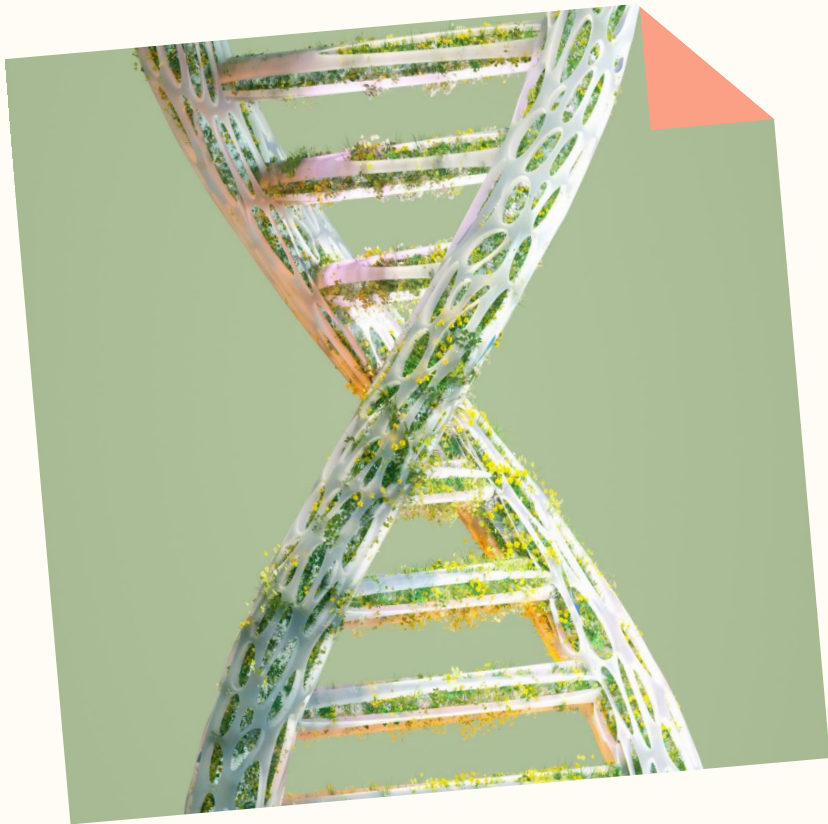
(UUIDs)

"Profiles"

Services & Characteristics

- Services
 - Basic device functionality
 - Groups of characteristics by function
 - "Generic Access", "Device Information"
 - "Battery Services", "Apple Notification Center Service"
- Characteristics
 - UUID & Value pair
 - Read, Write, Notify, Indicate
 - "Manufacturer Name String"
 - "Model Number String"
- UUIDs
 - Random number
 - Many are assigned by the Bluetooth SIG
 - Vendor / Service / Characteristic





Characteristic Operations

- Read
- Write
- Notify
- Indicate

Properties & Permissions

Security

Modes & Levels

Security Mode 1

- Level 1: No security (plaintext, no auth, no encryption)
- Level 2: Encryption + unauthenticated pairing
- Level 3: Encryption + authenticated pairing
- Level 4: Encryption + authenticated pairing + LE Secure Connections

Security Mode 2: data signing

Security Mode 3: new

- Isochronous broadcasts

Security

Pairing vs Bonding

Pairing: generating and using keys to encrypt the connection

Bonding: storing those keys so future communications between devices can be re-encrypted

“Legacy” vs “LE Secure” pairing

- Just Works (bad)
- Passkey (sniffable in Legacy)
- OOB
- Numeric Comparison (added in LE Secure)

Security How To

If you're worried about	use
Tracking	Resolvable random private addresses (plus IRK)
Sniffing	LE Secure Connections, encrypt communication
MitM	Authenticate devices (no Just Works)


Allowlisting devices is also possible

How Hack?

Tools

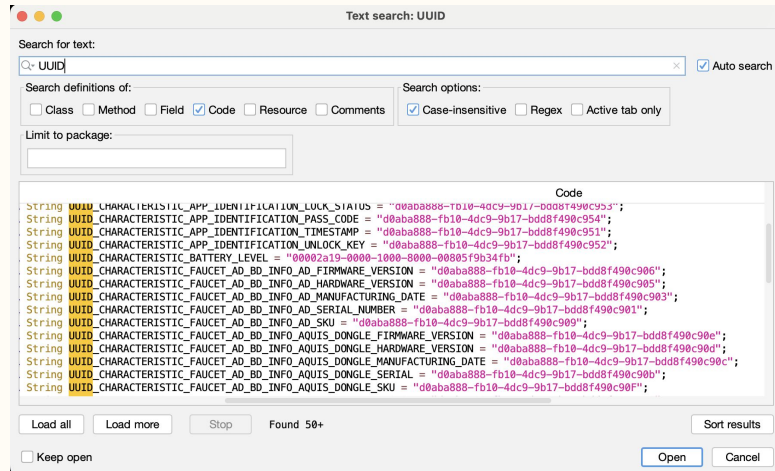
- NRF Connect app (ios/android)
- hcitool / gatttool (linux)
- bluepy (python)
- Nrf52840 + nrf connect desktop
- WebBluetooth! (experimental, maybe not supported anymore)
- CircuitPython boards!
- Sniffers
 - Nrf52 dongle (~\$10)
 - Ubertooth
 - Tcpdump
 - Android, macOS bluetooth debug

Scan the device

- Look at the services and characteristics.
- Google one or two of the UUIDs to see if they're known
- Read characteristics
 - Device name, vendor, etc
 - Other interesting data?
-  write to characteristics
 - If you own the device
 - (don't do evil, don't do crimes)

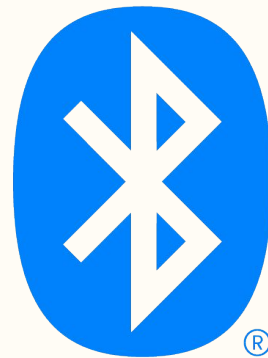
Reverse the App

- Is there an app?
 - Dumb question of course there is
 - What functionality is there? Play around.
 - Use jadx to reverse the apk
 - Search for "UUIDs"
 - What do they do?
 - Can you find strings related to features you found before?
 - (IANA iphone app guy)



Write to the device

- Did you find anything via reversing? Test it out.
 - Pair with the device if you need to!
 - Maybe just write stuff anyway (you own the device, don't do crimes)
 - Try to write to the UART Rx while Notify on UART Tx
 - You can do all of this with your phone
-
- Does the device have DFU open?
 - Can you make a custom firmware?



Interesting Vulns

Exploits?

VandyVape CVE-2019-16518

Sloan Faucets CVE-2021-20107

Owl Labs Meeting Owl
CVE-2022-31459, CVE-2022-31460,
CVE-2022-31461, CVE-2022-31462,
CVE-2022-31463

Various lower level protocol vulns



DFU Quick Note

DFU: Direct Firmware Update

From NRF's "Buttonless Secure DFU Service" doc:

"Configuring the Buttonless Secure DFU service to not support bonds does not restrict write access to the buttonless service. Any device will be able to write to the Buttonless DFU characteristic to enter DFU mode."

(still probably needs signature verification)



Demo

**Every BLE device is 1 hop away from
a compromised laptop**



Questions?